October 1978
Vol. 1, No. 9
ISSN: 0160-9572

The Seybold Report on Word Processing

IN OUR LAST ISSUE on "intelligent" typewriters, we wrote: "It will be a long time before people in offices begin to realize that it is not necessary to have a 'printer' at every desk. When that time comes, we can all have video terminals instead and we will be able to perform much more powerful information retrieval and manipulation functions than would ever be possible with even the smartest of intelligent typewriters." In this issue, we decided to elaborate on that statement by describing a system that is available today and which currently provides many of the capabilities to which we were referring last month. Actually, the Augment system from Tymshare is not as much a "system" as it is a capability or even a philosophy based on the assumption that people should be able to greatly "augment" their creativity and productivity, given the proper electronic environment and tools with which to work.

Although we will be discussing the Augment system, this article is not *about* that system alone. Its purpose is to provide some food for thought and perhaps to stimulate discussion on a topic that concerns all of us in the industry (users, consultants, vendors and philosophers), and that is, of course, *where are we heading?* Therefore, in addition to describing the Augment system and the philosophy behind it, we will be branching to related topics or issues and presenting other insights. In order not to disrupt the flow of the narrative, and so that those of you who are only interested in finding out about the Tymshare system *per se* will not be disturbed by these digressions, we have set them aside as boxes throughout the article.

The office of the future. When we think about office automation and the direction in which it is taking us (or we are taking it), we have little difficulty visualizing an environment in which video terminals abound, where document preparation becomes a "breeze," and where, instead of printing and mailing those documents, we will be able to distribute them electronically. And yet, this scenario seems so far away to most of us that it is hard to imagine exactly how such a system would look and feel. We know what it is like to input, edit and format letters, memos and longer documents at a display-based word processing station. Some of us have had experience transmitting information via communications from one word processing center to another, and some have even been successful in sending and receiving information to and from a large computer where it is miraculously absorbed into a data base of information. But there is another dimension to the "electronic office" that most of us have never experienced, and that has to do with the more dynamic process of information sharing and retrieval in a collaborative environment. This is where Augment excels, and it is for this reason that we have chosen to describe the Tymshare system, for it provides a working model of that aspect of office automation hardest to fathom—the relationship between a large and sophisticated system and the creative processes of its users.

An early version of the Augment system in use.

TYMSHARE'S AUGMENT
Heralding a New Era

We would like to make it clear from the outset that Augment is totally different from every other word processing system we have discussed so far this year. Most of our articles have described word processors that are intended to be used as correspondence or document production tools, and whose principal operators are "administrative support" personnel—secretaries, clerk-typists or WP operators. The Augment system, on the other hand, is designed for a different group of users with another set of tasks to perform. Many Augment customers are writers, scientists and programmers. Very often a group of geographically dispersed researchers who are working in related fields, or on the same project, will share their findings and comments with one another through the Augment system. The operators who use the system are generally not secretaries, they are professionals and technicians who are members of a common (electronic) community. For these professionals, the Augment system is not viewed as a tool to avoid the duplication of effort, but as a technique, an environment which moves them closer to their goals through the augmentation of the human intellect.

The Newspaper as Role Model

Many of our readers may not be aware that the "office of the future" concepts we all think of as being achievable in five or even ten years from now have been in actual practice for almost ten years in the newspaper and general publishing industry. Did you know that most of the daily newspapers in this country (and many of the weeklies) are produced electronically? Reporters take notes, conduct telephone interviews and write stories using video terminals. Those stories are routed automatically to other reporters for comment or collaboration, to a section editor for review and final editing, and then to a page layout editor who makes the final decisions about where the story is to be located and how much of it will actually go into the paper. In addition to the automatic electronic routing of copy from one person to another, these newspaper systems also incorporate levels of file security protection so that a cub reporter cannot edit his editor's final copy (although he may be allowed to see it), and so that the gentleman working on the sports section, or the lady writing a society column cannot access or gum up the international news department's stories, and so on.

Of course, newspapers are not novices when it comes to remote communications, either. Not only do they all receive several national newswires, whose contents are quickly and automatically incorporated into the paper's central data base of information, many papers also have remote bureaus set up in locations like Washington, D.C., or their state capital, where reporters also write their stories using video terminals and then transmit the news over dial-up telephone lines to the central office. There are even portable video terminals, for use by roving reporters or those who cover major sporting events or political conventions. Some of these terminals are so compact that they can slide under your seat on an airplane.

Incidentally, the electronic systems used in newsrooms (and in magazine and book publishing houses) around the world are not only being used to write, edit and route information hither and yon, but they also perform the composition functions preparatory to typesetting automatically.

The electronic newsroom at the Baltimore Sun.

The Genesis of Augment

In the early 60's, Doug Engelbart, an electrical engineer who had been involved with advanced digital computer component design, began to think about the eventual impact of the video terminal on everyday life. Engelbart's thinking went beyond the idea of having terminals available to everyone. He saw the use of the CRT as a way to expand man's mind—the augmentation of the human intellect, as he called it.

Engelbart theorized that before the advent of writing, man's knowledge had been limited to that which he could experience directly or learn about from others verbally. Writing changed all of that, he said, by providing a new dimension to our learning experience. Engelbart felt that the widespread use of video terminals as writing tools would provide an analogous added dimension for the human mind.

The brick trick. Dr. Engelbart experienced some difficulty conveying his idea of intellectual augmentation to others because there were no interactive terminal systems he could use to illustrate his point. (This was in the early 60's, remember.) In one paper he wrote on the subject, Engelbart decided that although he could not provide a practical demonstration of the augmentation of the human intellect, he could furnish an example of the *dis*augmentation of the intellect. So he tied a pencil tightly to a brick and asked someone to write with it. The results were atrocious, of course, and Engelbart included them in the published version of his paper! If you give a person terrible tools with which to work, he pointed out, the results will be justifiably mediocre; however, if you augment his capabilities by giving him superb tools, the results will be equally dazzling.

Enter SRI. Engelbart soon joined Stanford Research Institute in Menlo Park, California, where he launched his project in a special workshop called the Augmentation Research Center. Among Engelbart's early co-workers were Richard Watson, James Norton and William English.[1] The ARC project was funded by a series of grants from various sources, mostly from agencies of the Department of Defense. Engelbart's system was first implemented for display use on a small minicomputer, and consisted of the computer (a CDC 3100), disc drive and a video terminal and keyboard with a couple of added features —the keyset (or chord, as some prefer to call it), and a "mouse" (a movable cursor control device to be described in more detail later). The system was christened NLS[2] which stood for oN-Line System. The original NLS had only one video terminal, it soon grew to include a number of video terminals and several typewriter terminals.

The first big public demonstration of NLS took place in San Francisco at the Fall Joint Computer Conference in 1968. Engelbart set the terminal[3] on the stage of the Civic Center, and as he demonstrated the system a large-scale TV projection system captured the demo and displayed it on a huge screen behind him. It was a true multi-media presentation, in which images from the terminal display were juxtaposed or overlaid on the projection with various views of Engelbart's hands on the controls, and shots of his face as he described what he was doing.

1. English eventually spun off from the group and went to work at the nearby Xerox research facility (PARC—Palo Alto Research Project).
2. Augment is the name of the version of NLS that is commercially available.
3. The computer remained at the ARC lab. Engelbart's terminal was connected to the SRI computer via a dial-up telephone link.

Above: Douglas C. Engelbart, inventor of NLS/Augment, demonstrates the use of the system on an early model in his office at the Augmentation Research Center.

Below: Researchers experiment with the use of NLS/Augment in a conference environment. The image from the terminal screen is projected on the wall for all to see.

The NLS philosophy. The concept behind the NLS system is that each terminal should function as a person's *total* workstation. In other words, it is not simply an input and editing tool, it is a complete "knowledge workshop" which handles everything from initial note-taking through writing, editing, collaborating, revising, communicating, and finally, typesetting. It includes the ability to support graphics terminals for computer-aided design functions and for generating line art to be included in a document. It can function as a programming tool; it can be used as a calculator; it can access data bases and contact other systems; and finally, it can be used as an electronic mailbox.

Apart from its ability to facilitate the production of documents for broad distribution, the NLS system is designed to operate in a virtually paperless mode. Since all of the users on the system can communicate with each other via their terminals, they have little need for paper copies of most documents except for those that are to be shared with people who are not members of the NLS community.

NLS was never designed to be an inexpensive system. Its purpose was to give each experimental user as much computer power as he needed at his fingertips. The researchers at the Augmentation Research Center were attempting to design a system that would simulate people's normal working environments as they would come to exist in 10 or 20 years, when the cost of computing dropped. What happened, however, is that "knowledge workers"—the kind of people for whom NLS was designed (writers, managers, planners, researchers, administrators and engineers) soon began using NLS, not on an experimental basis, but on a real one. It spread from the small research center to all of Stanford Research, and then, in 1974, NLS reached out over the ARPANET, across the country and around the world.

The ARPA connection. The Department of Defense had a special organization, called the Advanced Research Projects Agency, which funded a number of technical projects throughout the 60's and 70's—many of which were not limited to military application. One such project was NLS. Another related endeavor, and one which has had an even greater impact, is the ARPANET. This is a computer-linking network that was set up to connect the computers at many of the major universities, research centers, and government facilities in this country and overseas. Starting in 1969, ARPANET was designed, installed, and maintained by Bolt, Beranek and Newman, the computer consulting firm in Boston. The network is presently supported by the Defense Communications Agency. ARPANET consists of a large number of minicomputers, called IMPS (Interface Message Processors) and TIPS (Terminal Interface Message Processors) which are linked together, and are, in turn, connected to the host computers at the various research and government centers. The entire network is designed in such a way that it is transparent to the user and is independent of the type of computer being used or its operating system environment.

An operator can log on to ARPANET from his terminal at one of the universities in Boston, for example, and, within seconds, could be using the PDP-20 at SRI International, or even hop across the ocean to run his job on a program resident in a London University computer. There are access security provisions, obviously, and not everyone can run a job on his choice of the ARPA computers anywhere in the world, but there is an amazing amount of flexibility and opportunity for resource-sharing.

If you have access to the ARPANET[4] you can use any one of several messaging facilities to either leave a "note" for someone else who uses the network, establish a direct terminal-to-terminal dialogue with him, or send him a copy of something you are working on. And, in addition to the special-purpose programs you may have available to you for a specific project, there are also a great number of general-purpose programs which you can use.

Electronic mail—here & now. There are approximately two hundred big mainframe computers on the ARPANET and another 65 or so IMPS and TIPS. There are several thousand people all over the world who use the facilities of the ARPANET on a daily basis. And there are probably many more thousands

4. In order to be a legitimate user of the ARPANET, you must be a member of a group that is working under a Defense Department contract.

The Electronic Underground

There is a large and growing network of home terminal users who devote some portion of their waking hours to using other people's computers, and/or "talking" to one another cross-country over established networks. Many of these people are, of course, college students, or friends of college students, or former college students, or friends of former college students, who have had access to university computers. Others are professional programmers who moonlight (or play) on their own terminals in their spare time, while still others, secondary school students or their parents, are simply computer "hackers." There are literally thousands of people who go home at night and log onto a system.

The most amazing thing is that many of these system users do not have computers of their own. They simply find a way to log onto someone else's. If you buy a computer terminal at your local personal computer store, chances are that a friendly salesman, anxious to clinch the deal, may hand you a phone number. It will turn out to be a dial-up line into a communications port on somebody's computer network.

When we logged onto the ARPANET with the assistance of a "legitimate" user, we decided to see if any high school students had been on the NET recently. So we asked the system, "Who is FRODO?*" And, sure enough, there was a record of a high school student from Atlanta, Georgia who had used that ID when logging onto the ARPANET a month or so previously.

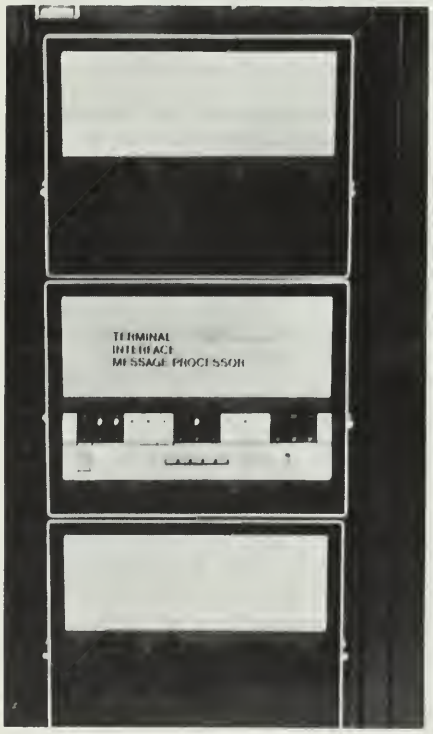
Once you have gained access to a system or network, it is sometimes possible to piggyback on other people's programs or to run some of your own, as long as you don't become too noticeable to the legitimate users by using up a lot of core memory or trying to pre-empt disk space. If you do become obnoxious, or if there are several of you on the same computer at once, chances are you will be bumped off the system by a legitimate user.

*There are certain names that are apparently popular with the under-18 set as user-ID's. Many of the characters from the Tolkien books may be found lurking in the shadows of a computer network, and you may even bump into "GOD" while you are exploring the depths of a system.

250	168	MCC-TIP	Refusing
251	169	NORSAR-TIP	Refusing
252	170	LONDON-TIP	Refusing
254	172	MIT-TIP	Refusing
256	174	LCSR-TIP	Refusing
257	175	WPAFB-TIP	Refusing
260	176	AFWL-TIP	Refusing
261	177	BBN-SPEECH11	NCP Down
262	178	DARCOM-TIP	Refusing
263	179	SRI-UNIX	Host Up
266	182	CHII	Random
303	195	NAC	Not Respond.n
305	197	BBN-TENEXA	Host Up
306	198	MIT-ML	Host Up
307	199	RAND-UNIX	Host Up
310	200	NADC	Random
312	202	LL-11	Random
316	206	CMU-CMMF	Host Up
320	208	AMES-11	Random
322	210	RADC-TOPS20	Host Up
323	211	NBS-UNIX	Host Up
324	212	ICEC	Host Up
327	215	USC-ECL	Host Up
332	218	PENT-UNIX	Random
347	231	SIAC-44	Host Up
354	236	MIT-MC	Host Up
356	238	NUSC	Not Respond.n
357	239	ILL-UNIX	Random
361	241	BBN-TENEX	Host Up
363	243	SRI-C3PO	NCP Down
364	244	USC-ISI8	Host Up

:KILL EAJ
243 163 ACCAT-TIP Refusing
244 164 CINCPAC-TIP Refusing
10-07-78 14:58:36 20 ED ARCTRN ED TTY

Above: A listing of a few of the computers on the ARPANET and their current status. Most of the descriptors include the name of the operating system—TENEX, UNIX, C3PO (!), etc.—the computer is currently running.
Below: A TIP on the ARPANET.



Here are a few characters we ran into on one of the ARPANET computers.

MSG: APT 1
DISTRIB: *AI, *DM, *MC, *ML
EXPIRES: 10/20/78 12:00:59
MHEM 10/06/78 12:00:58 Re: 1979 sublet
FULLY FURNISHED STUDIO APARTMENT AVAILABLE
JANUARY 1 TO JUNE 30. LARGE LIVING ROOM,
SMALL ALCOVE, SEPARATE KITCHEN (MODERN),
BATHROOM, HALLWAY, LOTS OF CLOSET SPACE.
HIGH CEILINGS, WOOD FLOOR, NICE BUILDING,
CLOSE TO M.I.T., IDEAL FOR VISITING PERSON.
$300/MONTH. CALL MARGIE 3-5847

Above: Electronic mail can be used for a variety of purposes.

Below: An interesting exchange of correspondence on the ARPANET about a famous chess match between David Levy and a computer.

MSG: CHESS 1
DISTRIB: *AI
EXPIRES: 10/14/78 09:26:38
GREENMIT-AI 10/06/78 09:26:38 Re: David Levy Letter
The following is a letter to Seymour Papert from David Levy.

"I do not know whether or not the fact was reported in the Boston newspaper, but I challenge match for our bet. The match took place in Toronto, at the Canadian National Exhibition, beginning on August 26th. It was a six game match but after I had scored 4 points from the first five games the sixth game became moot, and was not played.

I think it quite remarkable that you and I were both wrong in our predictions (in a way). You were absolutely certain that I would lose the match within 5 years, not 10, while I was equally certain that no chess program would be remotely near my strength by 1978 and I did not think that it would even be necessary to play a match. In the end the situation turned out to be somewhere in the middle, though happily for me it was on my side of the fence.

I was sorry not to have had the chance to meet you again when I played Mephisto on August 23rd. I hope that I shall be hearing from you shortly! Donald Michie has already paid his 500 pounds shared of the bet and John McCarthy's 250 pounds arrived yesterday.

Yours sincerely,

David Levy"

of users who can gain legitimate access to the computer-sharing facility. And, finally, the set of all the potential users (those who can arrange to log onto the network) is virtually unlimited! All of the current network users have available to them right now, the kind of computer power and ease of communications many of us can only dream about.

ARPANET users have had access to Engelbart's NLS system for about five years now. Some contractors for the Department of Defense have had NLS privileges as part of their D.O.D. funding. Other organizations have subscribed to the service with direct access over telephone lines. Four years ago, the subscription price for more or less full-time use of the system amounted to approximately $48,000 per year. Now that the cost of computing is coming down, the fee for the same access and usage privileges runs around $25,000.

Although it was getting cheaper, NLS was only easily available to ARPANET users (and, by extension, D.O.D. contractors). The people at the Augmentation Research Center felt that it would be appropriate to make NLS available to a much broader base of commercial and government users. But it obviously was not feasible to sell time on the ARPANET to non-authorized users or offer such a commercial service from a research institution. So SRI began looking for a commercial time-sharing company that had its own established communications network and which might be interested in NLS.

Tymshare Takes Over

And so it was that Tymshare purchased the rights to the NLS program in early 1978, committed to its ongoing development, and began moving software and staff from SRI to its own facilities in Cupertino, California. Doug Engelbart and James Norton joined the staff at Tymshare, as did all the rest of the ARC staff, plus a couple of long-time NLS-users who were prepared to start spreading the faith—Duane Stone from the Rome Air Development Center in New York and David Potter from the Educational Testing Service in Princeton.

Tymshare's arrangement with SRI allows SRI staff members to continue using NLS, but prohibits SRI from selling or permitting access to it by others on the ARPANET. Tymshare has taken over the NLS contracts for former SRI subscribers. Because the name NLS does not adequately communicate the product goals, Tymshare has renamed the new NLS program "Augment" and now provides Augment Service to both commercial and government people. The transition from SRI to Tymshare is now complete.

Tymshare was one of the original firms in the computer time-sharing services business, founded in 1965 by Thomas J. O'Rourke, still chairman and president of the corporation. It started as a small regional service company selling interactive computer services for engineering applications. As the time-sharing market matured, the company began to add business applications, especially financial analysis and data base management systems. At the same time, problems of servicing its customers over a vast geographic area led to the development of the first commercial value-added packet-switching network, TYMNET. At first, the network was used only by Tymshare to deliver its own time-sharing services, but soon other computers and users began to request connection to TYMNET, culminating in the formation of a subsidiary, Tymnet, Inc., licensed as a common carrier by the FCC[5] for packet-switching, message-switching, and electronic mail services to the public.

5. Federal Communications Commission.

Tymshare has now installed two of its own DECsystem-10 computers, called OFFICE-1 and OFFICE-2, on which to run Augment. It plans to add OFFICE-3 later this year. Commercial users gain access to Augment by subscribing to the service through Tymshare and "buying time" on the Tymnet communications network. Government users may subscribe to Augment and access it through Tymnet but generally access the Augment systems over the ARPANET at present. Tymshare is packaging the service for in-house installation on DECsystem-20's (2020 and up) with or without a connection to an external network.

The user interface. When customers subscribe to Augment, they may choose from among a variety of terminals. They may use any communicating typewriter terminal, such as a G.E. Terminet, or an LA36 from Digital Equipment, or even an IBM or A.B. Dick communicating mag card typewriter. They can also log on to the appropriate communications network via any standard communicating VDT that has both upper-and lower-case characters, such as a Hazeltine or Datapoint terminal. NLS was originally designed to be able to accommodate almost any terminal device a user might have or might want to install, and Augment has retained this flexibility. Although Tymshare recognizes that a typewriter terminal (particularly a portable one, like the Texas Instruments Silent 700) may be useful for occasional use, or for people who want to have a second terminal at home, the company strongly suggests the terminal that was developed by Engelbart *et al.*

The Augment 1200

Tymshare acquires its terminals from Data Media and reconfigures them into complete Augment workstations. These are, for the most part, conventional-looking CRT's that are derived from a standard off-the-shelf design. They have built-in microprocessors, (Intel 8080's) and are programmed by Tymshare to support the mouse and keyset peripherals. The communications hook-up to Tymnet is generally accomplished by means of a standard modem[6] arrangement, generally at a rate of 1200 baud,[7] although higher speeds are available.

The keyboard has a standard typewriter layout with a row of function keys above the main keybank. All Augment commands can be issued from the regular terminal keyboard (or from the keyboard of virtually any typewriter or display terminal. Most of the commands are entirely mnemonic in nature; for example, *i* and then *w* means "insert word." However, if you

6. A MODulator-DEModulator (sometimes called a Data Set or an Acoustic coupler) serves to modulate the digital signals coming from a computer or terminal into analog signals that can be easily carried over telephone lines that were designed to carry voice communications, and then modulates the data back into digital form. Many of our communications networks, whether they consist of above-ground or underground phone lines or microwave satellite carriers, or some combination of the two, are specially designed to carry digital information. However, until direct digital transmission of information becomes commonplace, we still need to use modems to handle that part of the journey from one location to another which takes place over ordinary phone lines.

7. The baud rate is roughly equivalent to the number of bits per second that are transmitted at one time. Actually the baud rate (named for a man named Baudot) represents the number of times the bit stream containing the information changes from 0 to 1. There are often 8 bits to a character, although this varies depending on the error checking information and the actual code structure being used. A rough rule of thumb for figuring out how much information you can transmit at a given rate is to divide the quoted baud rate by 10. Therefore, with a baud rate of 9600, you can transmit almost 1,000 characters a second. At 1200 baud, you would be sending information at approximately 120 characters per second, and so on.

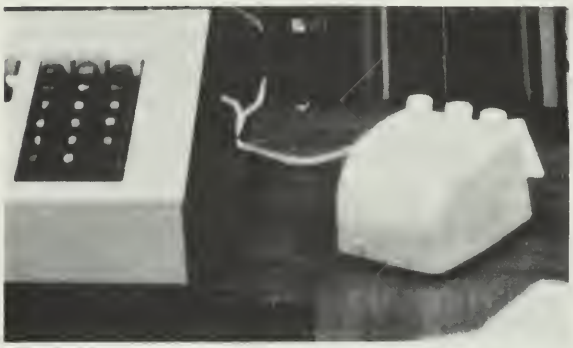
Right: David Potter, an experienced Augment user, editing text at the workstation. Notice that he is using the mouse & keyset, rather than the keyboard.

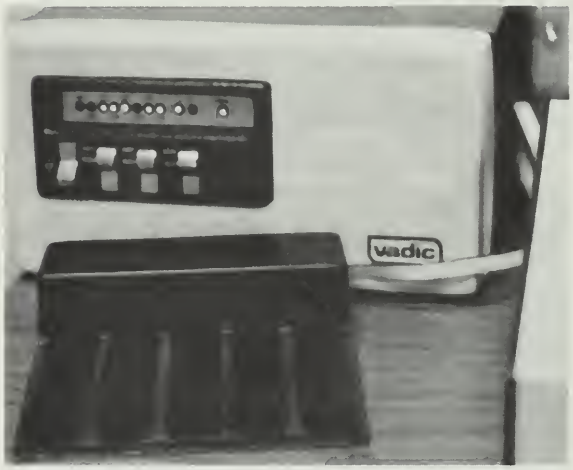
Below are close-up views of the mouse (top) and the keyset (bottom).

really want to use the Augment system efficiently, you will want to use the mouse and keyset.

The mouse, as we have mentioned before, is a cursor-control device, and it is also used for giving commands to the system (such as *OK, backspace word, backspace character*). The mouse is a small, hand-sized plastic device with three buttons on top. It is cable-connected to the terminal and rolls freely around on any flat surface. (There are two metal wheels underneath.) As you roll the mouse on the flat table beside the terminal, it will cause the cursor on the display to move. Once you get used to it, the mouse technique makes it easy for you to position the cursor more rapidly and flexibly than with a pad of cursor-directional arrows because you are not limited to four directions, but can move about at any angle. Once you have positioned the cursor under the character or point on the display you want to address, you can communicate that reference to the system and initiate any one of several editing functions by pressing buttons on the top of the mouse.

The keyset is a small device with five piano-like keys that is also cable-connected to the terminal. (Right-handed users usually place the mouse on the right and the keyset on the left. Left-handed users would probably want them the other way around.) Each time you press down one or more of the five keys on the keyset, you generate a character (letter, number or punctuation). The keyset works in binary code, so if you press down only the key that is farthest to the right, you will generate the following five-bit character: "00001," which translates to "a." If you depress the left-most and the right-most key simultaneously, generating "10001," you will get a "q" on the terminal screen. The total number of characters that can be generated in this fashion is 31. In order to increase that character set to encompass a 93-character repertoire, you can enter shift and supershift commands via the mouse. It is therefore possible to access the lower-case alphabet with the keyset in normal or unshift position, to generate upper-case characters in shift mode, and to get special or "pi" characters as well as numbers in supershift. As we mentioned earlier, all of the characters or commands which can be entered via the keyset or by using a combination of mouse and keyset, may also be entered at the teminal keyboard.





The keyset commands are quite difficult to learn. Once mastered, however, the mouse and keyset are the most efficient way to use the system. It isn't necessary to "think binary" but simply to develop a "chord striking" habit. If you watch an experienced Augment user working at the 1200 terminal, you will find that he rarely uses the keyboard except for entering long strings of text. Tymshare suggests that the new user famil-

iarize himself with the Augment command structure and become comfortable with the operation of the mouse before worrying about adding the use of the keyset to his bag of tricks.

The display. The Augment terminal has a display buffer of 8,000 characters which allows the user to save a screenful of information in local memory while he works on a subsequent screenful or to temporarily stash away the page he is editing while he checks another file or directory for related information. The user can also load in a program to double his screen capacity in either the horizontal or vertical direction. In other words, you can work in a regular screen-at-a-time mode; or in a horizontal mode in which you input and edit lines of information that are up to 160 characters long; or you can use the vertical mode, which allows you up to 48 lines of normal (80-character) width.

The terminal display itself is normally divided into two areas. The top four lines are for file, command and status information. The ensuing 20 lines are used to display text which has previously been input. As you enter new text it will appear on the screen from the top down, and the subsequent text on the display (which is actually the copy you have entered previously) will wrap down and "fall" off the bottom of the display. The display can be divided into up to eight different mini-displays, so that the operator may thus view and edit a number of different sections of a file, or a number of different files, simultaneously. And the user also has a choice as to whether he would rather see information displayed with light characters on a dark background, or dark characters on a light background.

Using the System

Augment's input and editing techniques are quite different from those found on other video terminal text editing systems. This is not due so much to the unusual mouse and keyset peripherals, but has to do instead with a number of other factors. For one thing, the original NLS system was designed to accommodate any kind of terminal the user happened to have. This meant that it was necessary to be able to communicate with the system (and to pinpoint locations and specify operations) using regular alphanumeric characters on *any* standard keyboard layout, and even without benefit of cursor controls. Moreover, the original editing package for NLS/Augment was written before most other video terminal text editors were conceived. Many of the terminal-editing operations we think of as *conventions* now, simply didn't exist then. The system designers invented their own solutions. Finally, the entire NLS/Augment system is based upon the concept of hierarchically-structured text, which leaves its own unmistakable stamp on the command language.

Structured files. Engelbart was convinced that a serially-oriented line-editing scheme[8] was not the ideal way to handle textual material. People don't write in chunks of consecutively numbered lines. They think (and write) in clumps and bursts, often jumping from one idea to another, sometimes retracing their steps. But always, Engelbart felt, there was an inherent logic or structure of ideas. If that intrinsic structure could be captured, it would facilitate the flow of ideas and make subsequent editing, collaboration, revision and output cycles easier. So, rather than arranging and addressing his files by arbitrarily broken and numbered lines of text, Engelbart chose to structure files according to their basic logical elements, and to organize them in levels which reflected their relationship.

8. Like Bowne's Word/One or IBM's ATS.

Text is, of course, made up of characters, words and strings or phrases. Augment allows you to address and to operate upon those grammatical elements in much the same manner as you would on any other text editing system. You can delete a word or define a phrase by indicating its starting and ending points. You can then move it to another location. But the command structure of Augment also encourages you to think of each idea, whether it is a sentence or a paragraph, as an essential building block. These ideas are called *statements*. A statement can consist of from one to 2,000 characters. Statements are related to each other in *levels*. Idea or statement 1 may have subsets 1A, 1B, and 1C, for example. These, too, may be sentences, phrases or paragraphs.

In addition to thinking of text or ideas in levels as in an outline form, you can also think of them as an organism, like a tree, that has a trunk (the overall subject matter or topic) and *branches*. You might have branches 1, 2, 3 and 4. Each branch would have its own "twigs" (1A, 1B, 1C, etc.) down to as many levels as are needed to reflect the structure of the information. Several adjacent branches and their sub-branches or "twigs" are thought of in the Augment system as a *group*, while all of the branches connected to a specific higher branch constitute a *plex*.[9]

Input and Editing

To start with, there is, of course, a procedure you must go through to log onto the Augment system and to create or call up a file. There are various commands and prompts that you interact with as you use the system. We will describe some of these briefly a little later on. However, at this point we want to discuss the basic terminal editing functions in the context of the hierarchical file structure we have just described. It is possible,

9. For those who have had any exposure to hierarchically-structured data base management systems, some of this terminology may sound familiar.

Hypertext

Ted Nelson, the author of a reality/fantasy computer "whole earth catalog" called *Computer Lib/Dream Machine** juxtaposed his description of Engelbart's hierarchically-structured text system with his own fantasies about how the ideal "video terminal as writer's tool" should work. Hypertext is Nelson's term for non-sequential writing, which, Nelson claims, is what we all do anyway. Rather than seeking to impose (or uncover) the inherent structure in our prose, Nelson feels we should let ourselves go—write (and read) with abandon, leaving links or pointers to related subjects which the reader may choose to follow or not, at his descretion.

"A *grand hypertext*," Nelson wrote,† "would be a hypertext consisting of 'everything' written about a subject, or vaguely relevant to it, tied together by editors . . . in which you may read *in all the directions you wish to pursue*. There can be alternative pathways for people who think different ways. . . ." "In my view," Nelson wrote in 1965, "a new day is dawning. Computer storage and screen display mean that we no longer *have* to do things in sequence; totally arbitrary structures are possible, and I think that after we've tried them enough people will see how desirable they are."

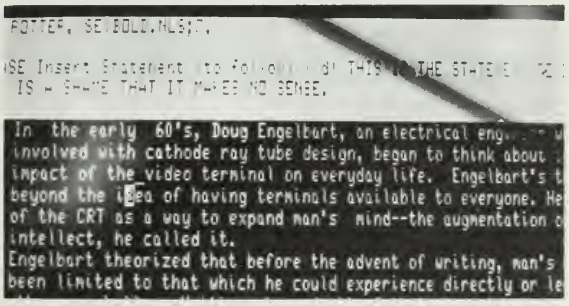
**Computer Lib/Dream Machine*, Theodore H. Nelson, 1974. Hugo's Book Service, Box 2622, Chicago, Illinois 60690.
†*Dream Machine*, page 45.

as we indicated above, to ignore the file structure and simply to input text in a continuous fashion. Or you can specify levels and statements as you go along. If you choose to do the latter you would end each statement with a command sequence (generally input simply by pressing a mouse button) to tell the system that you have concluded a statement. You might then indicate whether it should be considered to be on the same level as the preceding statement or whether it is to be one level below, or one or more levels above, its predecessor.

Augment will assume that the statement you have just entered is on the same level as the statement which preceded it unless you tell it otherwise. If the newly-entered text is to be considered subordinate to the preceding material, you would tell the system that it is one level below the previous statement by entering a "*d*" (for "down"). If, on the other hand, the new statement is to be placed one or more levels above its predecessor in the hierarchy, you would enter as many "*u*'s" as are needed to bring the statement to the desired level. (If you already had a statement 3A1 and wanted to make the statement you are about to enter a 4, you would specify "*u*," "*u*."

There is a single-character rubout command (the left-most mouse button) you can use as you enter text. And you can erase the last word you just typed by simultaneously depressing the center and left-most buttons on the mouse or by pressing the "backspace word" key on the terminal keyboard. The middle mouse button or the "command delete" key on the keyboard can be used to abort a command before you have confirmed it by issuing the "OK" command (the right-most mouse button). However, you will probably save all other editing or corrections until you have finished entering the file.

It is not quite as easy to correct typos (or to change your mind as you go along) when you are inputting on the Augment system as it is on many other cursor-oriented terminals. There is no simple way simply to run the cursor back through a line of text to overstrike or cancel. It is best to remain in "insert mode," and content yourself with rubbing out typos by backspacing on the fly. This is recommended procedure in any case, when your intent is to get as much copy into the system as possible. However, it is disappointing that on the Augment system, with its emphasis on nurturing the creative process, it is not easier to flip fluidly or transparently in and out of input and editing modes as on many other VDT systems.



Left: Inserting a statement. First you give the command "i,s" and then move the mouse to bug the location in response to the "noise words" (to follow), then type in the insertion. When you execute the command, the insertion will appear in place.

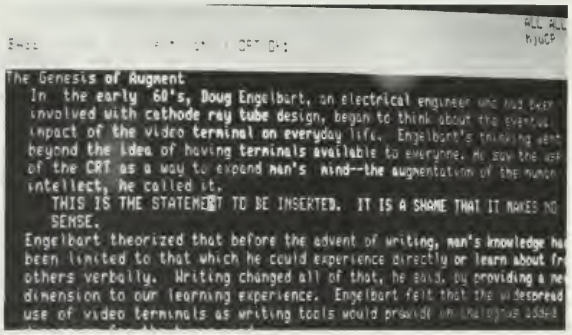
Right: To delete a statement you enter the "d, s" command, "bug" the statement, and, when you press "OK," it will be removed.

If you get lost or confused when learning to use the NLS/Augment system, you can ask for Help. Help is an on-line user's manual that provides a complete tutorial on every aspect of the system design.

Insertions. To insert text on the Augment system, you give a command which indicates both what you want to do and where you want to do it. In other words, you say "insert word" or "insert statement" or "insert text" (depending on which type of entry you wish to make). You must then tell the system *where* you want it inserted. The easiest way to do this, of course, is to use the mouse to "bug"[10] the statement which precedes that which is about to be inserted. This is the opposite of the way most terminal text editors work. On other systems you point to the text that will *follow* the information you are about to insert.

In addition to inserting text, you can replace characters, words, numbers, phrases, statements (or bigger units) simply by giving the command "replace word" (or character, or statement. . .) and then "bugging" the intended item. You then type in the replacement characters. You can use the same sequence to replace one item in the text with another one that is found somewhere else in the text by simply pointing to it as the replacement string instead of typing it in.

10. Augment uses the term "bug" to refer to the process of positioning the cursor at a specific location by moving the mouse until the cursor (a solid underline) appears beneath the character or space you want to address. You complete the "bug" by depressing the right-most button on the mouse to transmit the cursor location to the system.



You can copy a group of statements by "bugging" the beginning and end of the group you want to affect (or by identifying them by numbers) and then indicating via the mouse the location to which they are to be copied. Similarly, you can *move*, *transpose*, or *delete* groups of statements by using the appropriate command and then bugging the first and last statements in the group.

Search and replace may be implemented in a number of ways, depending on the user's preferences, needs, and level of sophistication in system use. For example, particular words or phrases can be located by using the command "jump to content (or word) next" and then typing in (or bugging) the desired text string. Augment will respond by displaying at the top of the screen the next occurrence of the desired word or phrase, which may then be edited as desired.

Alternatively, all occurrences of a particular word or phrase can be changed through Augment's *substitute* command. With this command, all instances of a user-specified word or phrase can be changed to some other word or phrase. As with all other Augment editing commands, users have the option of limiting the substitute operation to a specified portion of the document.

The Xerox ALTO System

According to a recent *Business Week* article,* both Tymshare and Xerox will be among the bidders hoping to supply the Executive Office of the White House with a total information processing system. Xerox has already installed a prototype of its ALTO system (so named because it is being developed at Xerox's Palo Alto Research Park) in the White House on an experimental basis.

We have not been invited to the White House to see ALTO, but we understand, from those who have seen the system and from Xerox's own "leaks" to the press, that it is a distributed processing system with minicomputer-based intelligent terminal subsystems. Not surprisingly (since some of the same people have worked on both systems), there appears to be a striking resemblance between the Augment/NLS system and the ALTO system. The Xerox system uses a mouse to control cursor movement and to initiate some commands. Terminal users will be able to communicate with each other and with a central data base of information through a telecommunications setup Xerox is calling the ETHERNET.

The resemblance between the two approaches apparently stops there, however. Whereas Tymshare is using a large mainframe computer to support many remote terminals, Xerox has a minicomputer in each VDT. The ALTO terminal has a full-page screen with bit-per-dot resolution enabling it to display complex formatted material (including line art) on the same terminal that is used for text input, editing and other forms of information processing. In addition to conventional daisy wheel and other printers, Xerox is also using a unique output device on the ALTO system—a direct output xerographic printer with a printing speed, according to *Business Week*, of 1,200 lines per minute.

*Communications section of the October 2, 1978 issue, page 40B. The article apparently is not included in all regional editions.

In addition, users can locate all statements satisfying given criteria (for example, all statements containing the word "glop" or the phrase "over the rainbow," or all statements changed since a given date and time, or all statements last modified by a particular user, etc.) by using the Augment command *set content (pattern to . . .)*" to institute what Tymshare calls a *content filter*, which may be turned on and off by the user. When the content filter is on, only statements possessing the specified characteristics will be displayed. Users can even apply these content filters to control which statements are operated on by Augment editing commands.

These editing operations are easiest to perform on the video terminal version of Augment. Although the typewriter version has all the capabilities of the display version, you cannot "bug" the characters, words, etc. on which you wish to operate, so you must identify them to the system by describing their location (by statement number, by a unique character string, or in any of several other ways). The typewriter terminal approach works, but it takes longer to implement the same functions. Because the typewriter stations are cheaper to rent or buy than the display workstations, however, you may want to use them for initial input of material or as portable or auxiliary terminal stations.

Other editing functions available to you on Augment include the ability to break and/or append statements, to transpose statements, to sort groups of statements, to edit tables as entities, and to change the upper- and lower-case status of previously input material.

One thing which you cannot do as smoothly with Augment's display editing tools as with some other systems is to scroll[11] through an entire file. In fact, even when you use the extended (8,000-character) buffer capability to let you work with up to 48 lines at once at the terminal, you do not scroll through the material, but you jump. To move the statement that is at the bottom of the screen to the top, for example, you would give a "jump" command and point to the bottom line with the cursor. The system would respond by jumping that statement (including all subsequent lines) to the top of the screen.

Communicating with the system. When you log into Augment with your own password, you will be rewarded with a terminal display full of information that you have asked the system to show you each time you log on. And that information will be presented in a format that you have specified. In other words, each user sees that which he wishes to see, and in the form in which it is most useful to him. Generally this preliminary information would include a file directory of the information currently residing in the user's own personal work area. It might also include information that has been transmitted to him by other users—either messages, or simply pointers or links to files in the on-line library or in their own file space.

After perusing this introductory information, you will probably want either to create a new file or edit an existing one. There is a default, or standard mode of operation within Augment called the *base* (for home base) subsystem. Most of your interaction with Augment will occur in the base mode. However, you can leave base at any time and enter any one of the other subsystems (e. g., sendmail, graphics, format, etc.).

11. Augment does allow the display user to "simulate" a typewriter terminal. By so doing, the user gains access to Augment's print command (not necessary in display mode), which can then be used to scroll automatically and continuously through an entire file or portion thereof.

Once you are in base mode, Augment will display (or print) BASE C:[12] and wait for you to enter a command to ask you what you would like to do. If you want to create a new file, for example, you would type (or input via the keyset) "C" for create. Augment will "echo" the command; *i.e.*, it will display (or print out) the entire word "create," and then prompt with a "C:." You type "F" and Augment again echoes the command by completing it, displaying "File." Augment then writes "T:" (meaning "text") and waits for you to type in the "text" that is the file name you have chosen. A file name may consist of up to 29 contiguous characters. It may not include spaces, commas, periods, or semicolons. Augment will respond by displaying a command line that gives your directory name (your Augment user name), the file name you have just entered, and the version number.

All communications with Augment take place in the same manner. If the display shows C:, you follow with a command—for most commands, just the first character of the operation you want to perform (*e.g.*, "I" for "insert," "D" for "delete," etc.) When Augment displays a "T:" you are to enter a text string, a file name or simply text. The system also makes use of "noise words" which are a form of innocuous prompting and which may be suppressed by the more experienced user. Here is an example of noise words as they would occur in the insert word command sequence: You see "BASE C:" and type "I;" the system echoes "insert" and adds "C:". You type "W." Augment echoes with "word" and adds the noise words "(to follow)" and the prompt "B/A" meaning "bug" the location or give an "address" (if you are not using a terminal with a mouse). You indicate the location to the system, and it will respond by prompting "T:" and await your insertion.

If you get lost while using Augment you can call for help at any time simply by typing an "h" followed by the command you want to have explained. The system will immediately branch to its own on-line user guide and display the information you need to continue. Generally this will consist of a discussion of the term you asked about and then a list or menu of further possible explanations. You can select an item from the menu by keying in its number or letter and Augment will respond by displaying more information about the item selected. All of this takes place without even closing the file you have open on the terminal screen. Once you have found what you want from the "help" subsystem, you can go back to "base" and the file you were working on simply by exiting the "help" subsystem.

Augment also offers other forms of on-line guidance. Any time you are not sure what to do next, you can simply type a "?" and the system will respond by giving you a list of all the options you have available to you at that juncture. In addition, users can easily access the complete on-line user documentation which is, of course, also available in hard-copy form.

Viewspecs. We mentioned earlier, in describing Augment's hierarchical file structure, that once you have allowed yourself to structure your input in the manner facilitated by Augment's command setup, you will have granted yourself a great deal of flexibility in the ways you can choose to review that input. This has to do with what Augment calls *viewspecs* (for "view specifications"), or the manner in which you specify that you want to look at a given file. A set of default viewspecs are automatically in effect when you log onto the system. These can be changed at any time. Viewspecs are related to the various levels in which the text was input. The standard or default viewspec (which can be changed by the user) shows all the levels of text, and all lines within each level. If you simply want to get an overview of the file, you might change the viewspecs to show only the first level, or the first few lines in each of the first level statements. Or you might want to zero in on a particular section of a file and ask for a detailed view of everything from a certain point on.

12. "C" stands for "command."

Sample Viewspecs

```
x - show one line and one level only
w - show all lines and all levels (default)
c - show all levels
d - show first level only
s - show all lines
t - show first lines only
a - show one level less
b - show one level more
r - show one line more
q - show one line less
m - statement numbers/SIDs on (default)
n - statement numbers/SIDs off
I - show SIDs not statement numbers
J - show statement numbers, not SIDs (default)
G - statement numbers/SIDs right
H - statement numbers/SIDs left (default)
y - blank lines between statement on
z - blank lines between statements off (default)
```

Shifting your viewspec in a document is akin to shifting your perspective from overview to detailed scrutiny with a single command. Using this very powerful capability, you can move in and out of the various layers of a file in a completely fluid fashion. In fact, this is one of the best approaches we have seen to the problem of viewing and digesting really long files of information at a video terminal.

Linking files. There is another capability that is inherent in Augment which is rarely (if ever) offered in other text editing systems. This is the *jump (to) link* command. At any time during the creation or editing of one file, you can jump to any other file in the system to which you have access. This allows you to integrate background information into a document you are creating, or to pull in related discussions or descriptions for your own reference which you may or may not want to include in your current document. It also allows you to work with two or more versions of the same document and to "jump link" from one to another in the process. When you jump, you can go to the beginning of the file, or to a particular character string or statement number. You can use the viewspec flexibility to find the location you are looking for, and then zero in on it.

If you combine Augment's various capabilities: linking one or more files together; shifting perspective, and dividing the terminal display into a number of independently controllable "windows," you have a very powerful "knowledge workshop" at your fingertips. You can then move and massage information, reference some items, copy or rewrite others, all while you are engaged in researching a topic, or revising documentation.

So far, most of the capabilities we have described have to do with the way in which *one* individual would use the Augment system. The power of the system is really unleashed, however, when collaborative work (research or writing involving two or more people) is involved.

The Augment Journal is, of course, very large. At the Augmentation Research Center where this picture was taken, they kept hard-copy duplicates of the Journal (the books on the shelves behind the terminal users).

Collaboration

One form of collaboration or communication between two co-workers who are physically separated from one another is the messaging facility. The simplest form of message transmission is actually completed outside of Augment. It can involve people using many different computers for their work. It is not even necessary for users to subscribe to Augment in order to be able to send messages to one another or around the world. As long as you have access to the ARPANET, you can use the "SNDMSG" facility developed by Bolt, Beranek and Newman as a part of the ARPANET contract (TYMNET customers instead have access to On-Tym). SNDMSG is an "executive level" program, which means that it runs as part of the Tenex or TOPS-20[13] operating systems.

SENDMESSAGE, the SNDMSG-like version embedded within Augment, can be used to transmit a message from one user to another (as long as both are known to the system as customers). By using SENDMESSAGE via Augment, you can send copies of the same message to a group of users by simply giving a single group ID. The message system will consult its user files, find that the indentification of the recipient you specified is actually a listing of a number of users, and dispatch copies of your message to each one. If you want to send several copies of the same message to a number of individuals who are not known to the system as a group, you would simply list each one as a recipient.

One advantage to communicating via terminal instead of calling someone on the phone is that if they are not in or are otherwise unavailable, the system will simply leave a message for them. The next time they log onto their terminal, they will be informed, "You have new mail." If you want to get into a long discussion you can simply leave a message and a phone number where you can be reached and let them call you.

Although you can send or leave messages, you can also communicate back and forth using direct terminal-to-terminal communications. You type "talk" or "link" and the name of the user you want to contact. If the recipient is logged on at the time, the system will just print out on his terminal, "LINK FROM (your name)." It works in much the same way as if you were working on a project in your office and someone walked in and said, "Hey, Joe. . . ." This facility serves as an aid to telephone communications. For example, users may establish terminal-to-terminal communication, and then, having established that both are there, they may continue their conversation on the telephone. Or they can go so far as to "link" their terminals together (using Augment's "shared screens" capability) and work on the same document, regardless of how far apart they may be physically located.

The Migration of the Office

Once true automation of the office starts, putting a terminal on every executive's desk, those people are going to want to expand their definition of "the office" to encompass a broader range of possible environments.

There are already thousands of people who work at least part-time from home terminals. We certainly expect that trend to continue, at least among the higher salaried "knowledge workers" in both government and commercial sectors. Of course, the luxury of staying home and logging onto the office network will only be afforded to those whose jobs do not involve the day-to-day tasks of managing people, supervising activities and on-site trouble-shooting. But the long-range planner, the programmer, the R&D person may very well be able to work just as effectively at home, or at his or her weekend retreat.

The status symbol for the next decade will therefore not be limited to the name plate on the door or the Bigelow on the floor, or even the terminal on the desk. No, it's the CRT in the study that will become the coveted distinction.

13. TENEX is the DECsystem-10 operating system developed by BB&N for the ARPANET applications, and the one under which Augment is normally run. Digital Equipment Corp. wrote its own version of TENEX for the big brother DECsystem-20 series (which is referred to as TWENEX by many programmers. Augment also runs under the TOPS-20 system.

Of course, it can be annoying to be interrupted in this fashion. For this reason some users will set up their own defaults that are the equivalent of saying, "Hold my calls—I don't want to be interrupted for a while." We understand, however, that most users are so intrigued with this messaging process that they do not mind being interrupted to carry on a dialog with someone in the same building or across the world. When a message or greeting suddenly appears on their screen, they will usually answer it, and their answer will show up on the sender's terminal screen. We have watched many such exchanges. In general they seem to be time-savers rather than time-wasters. If a question occurs to you while you are in the middle of something, and, rather than shelving it until it is convenient to call or write it down you send it through the system. You may get an immediate response, and save yourself a great deal of time and energy.

SENDMAIL is actually part of Augment's "journal" program. The Augment journal is a permanent record of all documents, notes or messages that have been "journalized" and marked with author-specified privacy or access restrictions. The journal program has two modes of operation. The "sendmail" mode allows you to transmit short or long documents to one or more users (including yourself), while the journal program *per se* will massage, sort, file and index that information for the permanent system record, or journal. Journal processing usually takes place at night, when the system is not busy servicing users. However, the sendmail portion of the program operates all the time.

To use the sendmail facility you first enter that subsystem and either respond to a series of prompts (about the title of your message and its recipients), or begin entering this information without prompts. If you do not know the IDENT or system name of the person, you can ask the system to find it for you, giving whatever information you know about him. Your message can be a a new file or a copy of a previously created file with comments. Some customers use this facility to extract a piece of a document they want to comment upon, send away that portion of the file along with their own suggestions. As with SENDMESSAGE you can route journal mail to an individual, or to several people. You can also send private messages to one or more people, and only they will be able to view or access it.

When you send mail to someone, you are not actually "physically" sending it, but simply giving him a link to that file residing in the common journal data base. Information about the file and who sent it will turn up in the recipient's work area file directory. He can then access the information directly.
The "journal file space" becomes a repository of all the documents and memos that were ever sent from one person to another using the sendmail facility. It therefore acts as an ideal project control mechanism, since all of the "correspondence" on a particular subject is resident in the journal file. Augment keeps a master copy of every file, and then as copies are made, and changes are made to those copies, it keeps track through the journal system of where those changes were, who made them, and when. All journal entries are automatically indexed and filed away by author, key title words and number.

Textual Data Base

Many of the word processing/information processing systems on sale or under development today anticipate a need for interactive access to information stored in a large file of textual data base material. Some on-line approaches automate the microfilm document storage process, reporting to the user the precise microfilm roll or fiche and the coordinates needed for manual retrieval. Other systems automatically find and display a microfilm enlargement on the terminal screen. Still other systems actually maintain the entire textual data base on-line in digital form, so that any portion can be called up on the terminal screen and viewed and manipulated in the same manner as any current document.

In whatever fashion information is stored, to be useful it must be capable of ready access. The simplest form of file retrieval is to present the user with a file index or directory from which he may determine and indicate the document he wishes to view. Some systems will not only "index" the file according to author, title, date and time, but will also include in a file header the keywords or topics used by the author.

The next level of complexity is exemplified by the New York *Times* Information Bank, in which the index consists not only of article titles, dates and authors, but is also intellectually indexed according to subject headings and thoroughly cross-referenced. As the information that is available to describe a file becomes more complex, so do access methods. A file which is identified a number of different ways is usually not subject to simple search by character string. The system will generally accept conditional search statements, such as "Show me everything written by Potter since Nov. 1, 1977 on the topic of "statistical analysis."

Ambitiously conceived textual data base programs offer a selective search on keywords not only of information in file headers or descriptors, but all or a portion of the text itself. The Data Courier System developed by the Louisville *Courier* allows the terminal operator to search on combinations of keywords that may be either in the story header or within the first 300 words or so of text. Criteria include the ability to restrict a search within a certain time frame and to use Boolean "and/or" combinations. Once the user has located the files he wants to see in this fashion, he must resort to a linked microfilm retrieval system to view the desired text.

A further variation is represented by the Mead Data Central information retrieval system, which is currently being used by the *Boston Globe*. It enables an on-line researcher to initiate a contextual search using a wide variety of text strings and criteria. Information within the data base has been inverted to facilitate rapid access. That is, a file locater for virtually all character strings has already been constructed. In this fashion, the program will examine all of the text in its files (consisting of everything that has been printed by the *Globe* (or set in type to be printed), retrieve the information, matching the criteria given, and display the actual text on the terminal screen, with keywords shown in context and highlighted (in color).

Finally, it is possible to store massive files without any particular index or guide (other than perhaps the general field of knowledge and chronology) and to conduct a search through the entire file for matching character strings. This apparently time-consuming and extravagant solution becomes more practical as data storage costs decline and with the use of an "intelligent" disk controller which only interrupts the central processor to report its "hits."

If you combine the sendmail and journal facilities with the editing and messaging capabilities we have already discussed, two co-workers or collaborators can access copies of the same original document and use Augment's shared screen facility to view each other's screens. They can take turns pointing out items, changing viewspecs and editing collaboratively. Alternatively, if two users split their screens into two or more windows, they can see their screen and the other person's simultaneously, and point to items on either side of the display.

Such cross-country collaboration has proven invaluable to many Augment users. Collaboration does not always involve document production, of course. Users who are working on other Augment subsystems may seek assistance for a variety of reasons.

Programming. There is an Augment subsystem called "DAD" ("Do-All-Debugger") that has been used successfully by a number of programmers. It apparently can be made to work for any of a number of programming languages from assembler to compiler level, and now works with the PDP-10 Assembler, Augment's L10 compiler, and the Jovial compiler. DAD uses the same general command set, mode of operation and structured format found in the regular (BASE) Augment system. It is therefore quite easy for programmers to create and edit their code on the system. They can use the SENDMESSAGE and sendmail facilities to confer with other programmers and/or ask for assistance. One major benefit to be derived from programming on the Augment system (apart from the collaboraton and editing capabilities) is that because of the nature of the Augment file structure, you are *ipso facto* engaged in structured programming.[14]

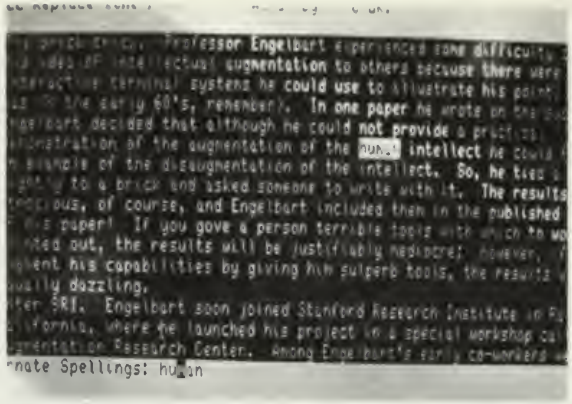
Calculator. There is a simple calculator subsystem that is roughly equivalent to a pocket calculator, except that it is online. It is a four-function, ten-digit memory program which allows you to "bug" numbers on the screen and specify the desired sequence of functions and then "bug" the location where you want the answer deposited.

Graphics. The Augment 1200 terminal allows the connection of a Tektronix 4014 or equivalent displays. And Augment offers a subsystem that will permit the input and editing of line graphics at this display station. In addition, once you have formatted your files for output to a typesetter or a computer output microfilm (COM) device, you can use the graphics terminal to view a "soft copy" representation of the proposed output appearance, a valuable aid in the pre-production stage.

14. Structured programming is a "buzz word" that has received a lot of attention in DP circles over the past few years. It has to do with the concept of writing code following a particular discipline. The theory is that if you write your programs in a structured fashion, you will avoid bugs. Some methods of structured programming also encourage the creation of documentation as part of the programming process.

The Adolescence of P-1

We heartily recommend this novel to anyone who is interested in programming, computers and/or telecommunications, or who just enjoys a fast-paced adventure story. Written by Thomas J. Ryan and published by MacMillan in 1977, *The Adolescence of P-1* is the story of a computer program that was developed by a student programmer and forgotten, and which suddenly turns up in a computer half-way across the country several years later, having migrated there on its own, through telecommunications. The story follows P-1 (the program) and its progenitor, Gregory, through a series of almost believable high-technology adventures.



Augment's interactive spelling checker flies through the text and stops to highlight any word it does not find in its dictionary—in this case, "humam." At the bottom of the screen Augment will display any alternate spellings it can find in the dictionary. If one of the ones listed is the word you *meant* to type, you can insert it by "bugging" that word.

Sort. In addition to the sorting capabilities which are available to you in the regular base subsystem, Augment offers a *sort* subsystem that can be used to sequence complex material, or for situations where you have a number of sort fields on which to operate simultaneously.

Spell. Augment provides a display-oriented, interactive spelling checker and corrector. It uses a standard 45,000-word dictionary, which can be supplemented by one or more unique user dictionaries. If the spelling checker cannot find a match for a word in any of the specified dictionaries, suggested spellings are listed for the user at the bottom of the screen. The user can then replace the misspelled word by "bugging" one in the list, by typing in a word, or he can choose to accept the word, either once or for the rest of the document being checked.

Tables. Augment will soon support the input, formatting and calculation of tabular information. Columns can be automatically justified right, left, center or aligned by decimal point. Rows or columns can be copied, multiplied, incremented, totalled or transposed with single commands. The user can place the *table* subsystem in an automatic mode. Columns and rows will automatically be re-summed when an element in the table is changed. And, of course, the tables so generated can be used as input to any of the other subsystems.

Forms. The user can create a replica of a printed form on the screen of his display by means of a "template." The template description includes specification of the layout, its size, and where the data is to come from, *i.e.*, from user input, the system (like dates and times), a procedure call, or from another Augment file. Options for linking forms together, to eliminate re-typing of information on related forms will also be available. The user can copy, output or send forms in the same manner that he would a straight text file.

Formatting. What you see on your screen is user-customizable and is generally related to the manner in which you have structured your text and to the viewspecs in effect. But the Augment 1200 system does not display underlined text, nor does it handle subscripts, superscripts or other pi characters gracefully.

Augment will format copy for an output device by using default values that are stored in the system, or by following mnemonic directives entered by the user or an Augment automatic formatting program. Tymshare writes formatting programs to suit the needs of particular documents such as manuals prepared according to military and government specifications. Both the default values and the directives are keyed to the various levels of text specified to the system. All first-level statements may be automatically treated as centered heads, for example, and a change in level can be associated with a change in indent condition, type style, or, for photocomposers, size or face of type. The Augment output processor will automatically number pages, generate headers and footers, and allow cast-off and specification of multi-column text material. It also enables the user to specify no-break blocks. At this point, Augment does not have automated footnoting. As noted above, Augment files may include line drawings. When output for an appropriate device, the drawings are delivered in place to the photocomposer for graphic composition.

Output. In addition to allowing the user to output documents on almost any type of character or line printer, local or remote, Augment can format and typeset documents for direct processing on photocomposition devices. Augment composes and justifies text in graphic arts fonts. Before sending files to a vendor, the user can preview typeset pages on a high-resolution Tektronix display. Although graphic arts fonts are not available and there is a limited size range, the user can verify line breaks and page layouts.

The Augment formatter-typesetter produces a general purpose, virtual phototypesetter file. Tymshare writes translation software to suit specific photocomposition devices. These include at present Information International's VideoComp 500, VideoComp 800, Comp 80 and Comp 80/2, plus the Singer 6000.

Hyphenation is performed (if requested by the user) by dictionary look-up. Augment does offer software to allow users to expand the dictionary to meet the requirements of a particular discipline.

Format as an Outgrowth of Structure

Even without Augment's hierarchically-structured text files, it is possible, on many systems, to infer the format the output document should take from the inherent characteristics of a text file. In tackling the conversion of "word processed" text to typeset form, most system designers rely on information that is already resident in the text file. In situations where word processing text is centered, or underlined, or indented or decimally aligned the conversion and/or formatting program will replicate those conditions in the typeset version of the copy. Many of these formatting situations are handled automatically on most systems. However, in some cases the user can redefine the form of the output at will, by changing the parameter specifications in a table of conversion values. This is the approach taken by Micom and by Wang, for example.

Another approach that is similar to the Augment concept, but which does not rely on the hierarchical structuring of text files, is found on the Atex system. There the terminal operators may differentiate portions of their text files through the use of up to eight different character display modes (light, normal, bold, reverse, and the underlined versions of each). For example, regular body copy might be entered in normal, major heads highlighted in bold, footnotes presented in reverse and underlining used for emphasis. When the file is actually composed for output, the system consults a table of format specifications which defines the typesetting parameters (type face, point size, line length, etc.) that are to correspond with each of the eight display faces *for that time only*. The same file can be output in an almost infinite variety of formats, simply by changing a single format file. One application in which this approach has proven particularly useful has been in the creation, revision and formulation of constantly changing insurance policies by word processing operators who know nothing about typesetting parameters. They simply work with the text files using the display modes and the system automatically hyphenates & justifies the text interactively according to the typesetting parameters that are "hidden" in the display modes.

Special Client Subsystems

With Augment's special programming environment, it is possible to construct specialized capabilities or subsystems for individual clients. One such subsystem, called the Financial Management System (FMS), was built for a government client. FMS addresses a problem area common to most government agencies namely, how to manage financial resources where funds are appropriated, and must be used in the fiscal year in which they are allotted. Priorities are constantly changing, and procurement dates cannot be determined ahead of time.

FMS has three modules, the *data entry subsystem* (DES), the *master file*, and a *user interface command language*. Each planned or on-going contract is maintained as an Augment statement in the master file. Down a level are all funding actions to date that pertain to the effort. A ledger file is also maintained, which notes every change made to the master file, providing a complete audit trail.

FMS users (managers, adminstrators and engineers) have a command language at their disposal similar to the others in Augment, but with unique terminology relevant to their everyday work. They can view the master file through special filters, search and copy parts to their own working files, print out specially formatted and totalled reports, or use the base, calculator, sendmail (etc.) subsystems to refine or enlarge the scope of the document.

Managers can play the "what if" game, making temporary changes in priorities, start dates of contracts, level of funding, etc. The ability to propose changes and to determine their financial effect before committing to a course of action has proven invaluable.

Pricing

Since Augment is a time-sharing service, pricing depends on the amount of time you spend using it, and on the manner in which it is used. It is therefore difficult to arrive at an "average" price. Because the cost of using Augment is made up of a number of components, including communications costs and terminal purchase or rental as well as Augment's own service charges, the easiest way in which to visualize the total picture is to break it down to an overall hourly charge.

If you are logged onto an Augment 1200 workstation and are engaged in text entry, manipulation and communication, it will cost you a little over $16 an hour. If you anticipate using an Aug-

The INtext System

Interactive Systems Corp in Santa Monica, California offers a documentation system, called INtext, which is based upon the UNIX operating system developed by Bell Labs. We described some of the system's capabilities in an earlier mention (see Vol. 1, No. 2, pp 15 & 16). Worthy of special notice here, however, is the fact that the company now has typesetting composition and output driver programs available for Mergenthaler's V-I-P phototypesetters and its new high-speed CRT typesetter, the Linotron 202. Programs for other typesetters are available on request. In addition to its regular composition program (called INroff), Interactive Systems also supports a math preprocessor (EQN) and a table formatter (TBL) which were both developed by Bell Labs. Now that the I/S system can compose and format multilevel mathematical equations, and can drive an output device that has a virtually unlimited character repertoire, it becomes a likely candidate for consideration by those users who need a documentation system that can handle complex equations as well as running text. INtext also offers a terminal-oriented text editor (INed) and an electronic messaging capability (INmail).

ment terminal full-time during a normal 40-hour work week, it would cost you more per *week* to use Augment than the *monthly* rental charge on most stand-alone word processing systems. Tymshare doesn't think you should look at it that way. After all, they claim, the average Augment customer probably only spends about two hours a day using his terminal, for a cost of $33.20 per day. Besides, they say (and they are right), you cannot really compare a "knowledge workshop" like Augment to a word processing system that is used as a production tool.

Summary and Conclusion

As you see from the pricing information above, Augment is not inexpensive. However, we know of no other system on the market today which provides the same broad range of information handling capability. True, there are other electronic mail systems and there are even a few systems that enable the user to input, edit and output documentation combining line art and text. But no design group has given more thought to the relationship of such a system to the human thought processes than the people at the Augmentation Research Center. We don't mean to imply that the Augment system is the easiest of all to learn, for it is not. Nor do we want to give you the impression that it is the type of system that can be efficiently used by most clerical or "typical" administrative support personnel. Augment belongs in the hands of the professional—the business executive, decision-maker, writer, scholar, engineer, or scientist—not their secretaries. Not that secretaries couldn't learn to use it, or that it wouldn't enhance their productivity. But the real power, the real genius behind the Augment approach has to do with the way in which it interacts with the creative, probing mind. It is a tool for people who are willing to become engrossed in the interactive process and who are eager to explore the open-ended possibilities such an opportunity affords.

Our reaction. Our major criticisms of Augment are: It is too expensive for the majority of users when compared to the cost of most of today's video terminal word processors; the system's basic editing features are not as easy to use and implement as those found on many other terminal text editors. And, because you are always working at arm's length from the computer (over a communications line), terminal response times can suffer, depending, of course, on the baud rate used and the load on the system. If one could access the same powerful resources on a cheaper system with a fast, intelligent interactive terminal that has its own local storage (to facilitate text editing and scrolling functions), then we would definitely be ready to stand in line.

We made the point early in this article that Augment doesn't much "care" what kind of terminal you use. It is very accommodating. It could even be a word processor with a telecommunications interface. In fact, we would settle for that kind of tool, without the mouse, without the keyset, but, hopefully, with virtual scrolling.

But if we had that kind of tool we wouldn't need to hang on to "big daddy" all the time. Lots of times the ordinary functions of the word processor would be just right.

But there may be times when we would like someone to look over our shoulder. "Oh-oh, you spelled it wrong." Or, "make it simple, stupid; you won't pass the Flesch test with those involuted sentences." Or, on our own part we might say, "Hey, big daddy, just where did George Boole teach symbolic logic?"

Distributed intelligence is the word for it. There can be tasks appropriate for various levels of intelligence, and there are very few of us who couldn't do with a little augmentation from time to time.

User reaction. Augment users are delighted with the system. We have not found anyone who would rather have waited until something "better" came along. Not only do they love the system, they can't help feeling that they belong to an elite community. It is stimulating to belong to the electronic intelligentsia.

The people at Tymshare will probably think this is a gross oversimplification, but we see an analogy between the HAM radio operator and the Augment experience. There are many amateur radio buffs who are engaged in stretching the technology, and there are others who are themselves stretched by their involvement. For those who use it on a regular basis, Augment is more than a "system," it is a way of life—an absorbing, enriching experience. One customer who had just resumed using Augment after a year of being without the system, told us, "I feel as if I have just been given back part of my mind!"

Patricia B. Seybold

Letters . . .

Word Processing: A Mushrooming Tiger?

I have recently become a subscriber to your monthly publication. The material and the format presented on various types of equipment by suppliers is excellent. However, I would like to see more in-depth coverage of many other aspects of installed Word Processing. In particular, such things as:

A. Office design
B. Job Enrichment
C. Centralization vs. decentralization
D. Clerical unions
E. Employee acceptance and conerns

From our studies, it appears all WP installations imply new job titles and functions. We believe it should be installed without increasing internal bureaucracy. That is, WP should not be restricted to specific resources. At the present it would appear WP will mushroom into creating another Data Processing tiger.

Your comments on these subjects would be most appreciated.

D. Schwartz
Information Systems
Cardinal Distributors Limited
5400 Hochelaga
Montreal, Quebec, H1N 1V9, Canada

A User's View of Vydec Systems

I am presently working as a free-lance word processing operator, and I thoroughly enjoyed the [Vol. 1, No. 4] issue's very detailed discussion of the features and operation of the Vydec Text Editors. I feel this makes The Seybold Report more useful than other publications which touch upon machine features but say little about what is actually involved in USING those features during document input, revision and printing.

I believe your equipment reports would be even more valuable if your discussions of the operational details of any particular machine were reviewed by the manufacturer and/or highly proficient operators of that machine. In this way, you could minimize inaccuracies and perhaps become aware of techniques for machine operation which are more efficient than those described in instruction manuals.

For example, the following comments could be made about your review of the Vydec 1200 and 1400:

Page WP-6, column 2, last paragraph: The statement is made that ". . . when you clear the screen to file the first page back on disk . . . ." Actually, storage of text is one thing; clearing the screen is a separate operation, so that it is not strictly correct to speak of clearing the screen to file text on a disk.

Page WP-8, column 1, last paragraph: The number which appears in the cursor position window when the cursor is at any particular horizontal position can be varied by means of the increment and decrement buttons. Therefore, the number does not necessarily correspond to the horizontal position of the cursor in the sense that with any particular (horizontal) cursor position, there will be associated one and only one number in the cursor position window. (For example, the number which will appear in the window when the cursor is in the "home" position can range from 1 to 255.) The increment and decrement buttons are used to set the left margin—to synchronize the horizontal cursor positions on the screen (as reflected in the window display) with the desired print positions on paper. It is only in this connection that the increment and decrement buttons help in moving the cursor to a specific location and setting tabs and margins. The buttons themselves are not used to actually reposition the cursor, and they are inoperative when the line-number option for the cursor position window has been activated.

Page WP-8, column 2, first paragraph: It is not correct that 1.5 line spacing can only be initiated at the control panel. 1.5 line spacing can also be effected by inserting a subscript code (two, if the Text Editor is equipped with ¼-line subscript and superscript codes) at the beginning of every line except the first. An editing program could do this very quickly, although it would be still faster to make use of the appropriate printer line-spacing selector button on the control panel. I might mention that the double-space key does not have exactly the same effect as the double-space button on the control panel.

Page WP-9, column 1, fourth paragraph: If a disk is being scanned in leisurely fashion, it will be necessary to clear the screen after reading one track and before reading another. Only when "thumbing through" (by holding down the read button of the desired file) is the screen automatically cleared between the reading of tracks.

Page WP-9, column 2, last paragraph: The instruction to "space or tab over to that point" should read "space or tab over to the right that many spaces, beginning at the left margin," since the result of the calculation does not necessarily coincide with the horizontal cursor position at which the cursor should be located before the heading is typed. A heading can also be centered "over a page" (without regard to the lengths of lines of text) by moving the cursor to the (horizontal) center of the paper and moving the cursor to the left one space for every two characters in the heading, and then typing the heading.

Page WP-10, column 1, paragraph 5: The statement is made that words may be underscored individually by striking the Brite Start and Brite Stop commands at the beginning and end of each word. Actually, the procedure for individually underscoring a word is to position the cursor at the first character of the word and then depress the Brite Start key. To complete the procedure, move the cursor to the immediate right of the last character of the word and depress the Brite Stop key.

Page WP-10, column 1, paragraph 6: The description of the save procedure is not quite correct. The first step is to place an end code at the end of the block of text to be saved. (This, however, is not necessary if there is no text on the screen following this block.) The next step is to move the cursor to the first character of the block. The last step is to depress the Save key—once if the Save track was previously erased, twice otherwise.

Page WP-10, column 2, first paragraph: Erasing the copy block from its original location can be accomplished much more quickly if nothing below the block on the screen or in the buffer is needed. In this case the cursor can be moved to the first character of the "unwanted" block and the Page End key depressed. This will erase the block and everything below it (including material in the buffer) in one stroke.

Page WP-10, column 2, paragraph 4: This is a needlessly cumbersome method for merging copy. Text stored on one track can be effortlessly inserted in the middle of text from another track, except when the insertion would entail exhaustion of the capacity of the screen + buffer. (In this case even the procedure described, or the save-and-recall process, might entail loss of some of the text at the end of memory, and more involved procedures would be required.)

Page WP-11, column 1, first paragraph: An essential step, depressing the Adjust Margin key, was omitted from the description of the procedure for block-indenting a paragraph. (This step should follow the resetting of the bell at the right edge of the screen.) The procedure can be made a bit more efficient. Efficiency can be improved even more if editing programs are used.

Vydec User's Views (continued)

Page WP-11, column 1, second paragraph: If the Text Editor stops for a hyphenation decision because a word exceeds the Hot Zone, the cursor will not stop at the end of the line but rather will stop to the immediate right of the right edge of the Hot Zone.
Page WP-11, column 1, third paragraph: In printing justified text containing block-indented paragraphs, the right margin/bell is reset whenever there is a change in the right margin/bell.
Page WP-11, column 1, fifth paragraph: Subscript and superscript codes (and coded functions in general, with several exceptions) appear as distinguishable characters on the screen only on Editors equipped with the Character Generator feature.
Page WP-12, column 1, third paragraph: It would be more accurate to say that the Brite Start and Code functions, in conjunction with function keys, are used to generate command characters. It is not correct to say that any function key can be used to generate command characters.
Page WP-12, column 1, seventh paragraph: Merging a mailing list with a form letter can be accomplished with programs. (Generally speaking, the full capabilities of the Vydec 1200 and 1400 can only be realized by operators who are proficient at constructing editing programs.)

Sincerely,

Kent Hirata
1835 Pine Street, #12
San Francisco, California 94109

—We appreciate your taking the time to respond in such detail to our Vydec article. You are right, we should have the accuracy of each article checked by the vendor, and we do make it a practice to do so. However, Vydec did not respond to our request for verification until after the article had gone to press, so, in this case, you are right, we missed out. We suggested to Vydec that they write a letter for publication detailing any points with which they disagreed. However, since they have not done so and you have, we are delighted to publish yours instead.

In our own defense, however, we would like to point out one thing: when we describe how a particular function is performed, it is only included as an illustration. We do not, nor could we, describe every detail about how each operation is performed, and, sometimes we streamline the description by omitting what seem to us to be rather trivial incidentals. Nevertheless, we are glad for your detailed feedback.

Greetings from Datapro

I would like to take the occasion of receiving your fifth issue—the one on the IBM OS/6 product—to tell you how much I have enjoyed reading the Seybold Report on Word Processing. The in-depth reports you do in the inimitable Seybold style are really very useful and I look forward to meeting more of them as you choose to review additional systems.

Best personal regards,

Amy D. Wohl, Editor
Datapro Reports on Word Processing
1805 Underwood Boulevard
Delran, New Jersey 08075

—Thank you for the testimonial, Amy. We enjoy reading your Datapro Reports, too. In fact, when anyone asks us what else they should be reading in this field, we always recommend Datapro, for it offers the kind of industry overview that is indispensable. We feel that the two publications are quite complementary—ours for an in-depth evaluation of one particular system at a time and yours for its checklist of features and prices across the entire industry.

THE WORD ON WP

IBM's DPD Announces WP System

IBM's Data Processing Division has just announced a whole series of distributed processing products, including a brand-new "office communications" system. The IBM 3730 distributed office communications system is capable of performing word processing and data processing functions concurrently, and can be operated as an independent, stand-alone cluster system or connected to a host IBM 370 mainframe computer.

The system is based on a new model of IBM's 3791 communications controller. Each 3791 can support up to 16 different input and output devices, including up to twelve 3732 display terminals and up to eight 55 cps Qume printers. Each 3730 cluster can be configured with from 5.5 to 24.1 million characters of disk storage. The 3730 clusters will operate as independent word and data processing systems. However, they increase in power and flexibility when connected on-line to a host 370 computer.

Pricing and availability. The purchase price for a 3791 controller and disk combination ranges from $32,660 to $44,860, depending on the amount of disk storage required. Charges for a two-year lease on the same equipment would run between $986 and $1,285 per month. Monthly rental charges run from $1,159 to $1,510. To the price of the controller must, of course, be added the cost of the terminals and printer.

IBM's 3732 display terminals sell for $3,325 each. Two-year lease on the VDT is $95/month, while rental costs are $112 per month. The daisy wheel printer costs $7,455 to purchase, leases for $213/month under a two-year contract, and is available at a monthly rental price of $250. Additional peripherals that will be optionally available on the 3730 system will include a sheet feed device for the daisy printer, a high-speed line printer, and IBM mag tape drive, and a special communications attachment that will allow the system to accept input from a communicating Mag Card/2 typewriter.

IBM has announced that it will begin marketing the 3730 system in selected cities (New York, Boston, Washington, D.C., Chicago, Dallas, Los Angeles and San Francisco) the first quarter of 1979. However, these will be "Phase 1" systems, meaning that they will provide basic word processing functions. The "Phase 2" versions of the system will include extended text processing software and the capability of attaching to the host 370 system. In other words, you can put in a 3730 cluster system next year and gain familiarity with the product and its capabilities, and then upgrade the following year to the on-line host attachment and expanded WP features.

Functions and features. We have not had an opportunity to try out the 3730 as a word processor yet, but we are told that it offers all the standard word processing functions you would expect to find—insert, delete, block move, automatic letter writing, etc. When the host connection is available, it will allow the terminal system to be interfaced through an SDLC communications protocol to any IBM

IBM's office of the future concept involves the use of distributed processing cluster systems for word & data processing, which can be connected on-line to a host 370 mainframe computer to be used for large-scale word processing, storage of textual data bases, & electronic distribution of documentation.

370 that operates in a virtual system environment. Once the data that is generated and massaged at the 3730 level is input to the 370 system, it can be handled in a number of ways by user-generated programs. Customers will be able to implement their own centralized electronic document filing, retrieval and distribution systems.

From what little we know of the system so far, it appears that IBM is on the right track. DPD customers with one or more 370 systems will be able to establish information processing networks, using the data base storage, retrieval and manipulation capabilities of the big computers, yet giving the user a lot more flexibility and power where it counts, locally. For example, the 3730 system could be used to prepare, edit and update massive quantities of documentation, using "state-of-the-art" interactive terminal editing functions, and then transmitting that information to an ATMS or SCRIPT program for final formatting.

The DDP/OPD relationship. Although the 3730 is clearly a DPD product since it is so closely related to the 370 product line, we were surprised that IBM would sanction DPD introduction of a word processing system by its Data Processing Division, which could be viewed as competition to the Office Products Division's own System 6. However, the people at DPD point out that the OS/6 is a stand-alone system, theirs is a cluster approach, and that DPD's involvement in word processing is just another example of the strong cooperative marketing effort that has been going on across group boundaries for the past year or so. OPD has apparently been aware of and involved in the 3730 product development, and has, in fact, allowed the Data Processing Division to hire many of its most experienced marketing support representatives and sales people to help them get started on the right foot.

Distributed Processing Products. In addition to the 3730 announcement, IBM introduced a "major new family of distributed processing products" called the 8100 system, which incorporates a new memory chip technology that contains 64,000 bits of information on a single computer chip. Another high-density memmory chip used in the 8100 product line offers 18,000 bits of storage on a single chip. In addition, the 8100 products use a new high-density removable floppy disk developed by IBM that will hold one million characters of information.

Another departure for IBM which coincides with these new product announcements is the fact that with the 8100 system, IBM has apparently broken with long-standing tradition and has "unbundled" hardware and software pricing for the 8100. Like the 3730 system, the 8100 can be configured as an independent cluster system (with one or two controllers), or attached on-line to a host 370 system.

Finally, IBM announced a distributed processing system configuration for industrial applications. Called the 3630 system, it is apparently analogous to the 3730 system in terms of hardware configuration. However, it has been "ruggedized" for use an an input and control system in an industrial (rather than an office) environment.

New Member of Wang's System 5 Family

The SICOB (word processing and computer equipment) show that was held in Paris in September was marked by the unveiling of a new Wang system configuration. Wang introduced the System 5, Model III, and announced a couple of enhancements to the Model I version as well.

As they were introduced at Syntopican last June, Wang's low-cost System 5 family consisted of the Model 1 (priced at $9,900), whose characteristics included all of the basic editing features found on most other Wang systems, except the block move, copy replace, glossary, merge and note functions, while the Model II (priced at $11,900) had all of the features Wang's higher priced 10A system had except the auxiliary glossary functions. Both models consisted of a single system controller which contained a microprocessor and dual floppy disk drives, a terminal and one of Wang's own daisy wheel printers.

The new System 5, Model III, uses the same hardware configuration and the same word processing software, but makes all of the auxiliary glossary functions available. These include paragraph numbering and retrieval, automatic footnoting and alternating page headers and footers. In addition, the decision processing capability is available to users who want to try their own hand at creating special-purpose "programs." Also included as part of the standard Model III software package will be the Math Support and Sort packages.

System 5, Model III, is priced at $13,900. As we predicted in our coverage of Wang's System 5 after Syntopican (*see Vol. 1, No. 6, p. 7*), the company will be discontinuing its earlier stand-alone system, the 10A, very shortly. What this means, of course, is that the remaining difference between the System 5 and the System 10 (the ability of the latter to support auxiliary input and output functions) will probably fall by the wayside as well. We would most certainly expect a communications option to be forthcoming shortly for one or more models of the System 5.

Another thing which may be of note is that Wang is now quoting separate pricing for the master station (terminal plus disk and system controller) and the printers. So, it may soon be possible to have two or more terminals sharing a common printer, which would bring the per-terminal cost down even more.

Amtext 425 keyboard

Amtext 425 Enhancements

Addressograph-Multigraph's full-page display word processing system is really coming into its own. AM is now shipping systems with the new, modified keyboard design which, by the way, bears a striking resemblance to the keyboard on AM's brand-new (and very powerful) typesetter, the Comp/Edit. In addition to simplifying and rationalizing the keyboard design on the 425, AM has also been working on a new software release which is said to be very exciting. Already released to the field are the system's automatic paragraph assembly package, a bold print mode, a local print mode which allows you to type directly from the terminal keyboard onto the Qume printer, and an expanded save/call capability. The paragraph assembly program will automatically retrieve numbered paragraphs from any documents on the floppy disk and assemble them, in the order specified, according to whatever format the user desires. You can have up to 53 documents on a disk, and each of those documents can contain up to 999 numbered blocks (paragraphs, sentences, words or phrases). However, the total is obviously dependent on the total floppy disk capacity. The save/call block feature allows you to store away (on disk) from one to four blocks of text for future (or repetitive) recall. The total amount of text that can be stored for recall in this manner is 9,984 characters.

Among the more exciting features expected to be released soon by AM are a complete column handling capability that will allow you to move columns of numbers around on the display, a very powerful sort program and a math package. In addition, AM has announced the availability of an asynchronous communications option. Bisynchronous communication is promised for release early next year.

WP Briefs . . .

Philips has introduced the overseas version of the Micom 2000 system under the name P5002. The product is essentially the same as the Micom system we reviewed last spring (see Vol. 1, No. 3). However, Philips Data Systems has packaged the product slightly differently, by building the floppy disk drives into a desk, and altering the terminal housing slightly. The new Philips product now looks very much like the old Philips/Norelco product line which, as we understand it, has been discontinued, at least in Europe. The P5002 comes in a number of different language versions, with different keyboards, display character sets (with accents) and prompts available for each major European language market.

Greyhound Computer Corp. of Phoenix, Arizona, has entered the word processing business by purchasing "an undisclosed quantity" of equipment from Artec, International, which it hopes to lease to end-users. The total value of the transaction was reported to be in excess of $3 million.

Hendrix and A.B. Dick have apparently signed an agreement whereby A.B. Dick will have the worldwide marketing rights to the word processing system that was developed by Hendrix Electronics. The WP system will consist of a controller and up to three full-page display terminals. The price is expected to be between $17,000 and $35,000, depending on the size of the configuration chosen. The Hendrix system is said to be compatible with A.B. Dick Magna I and Magna II equipment. The two companies are apparently still discussing the possibility of extending the agreement to include OCR equipment.

The Seybold Report on Word Processing. Published monthly, 12 issues per year. Subscription prices: U.S. $60 for one subscription, $48 each for additional copies sent in the same envelope. Special discount for subscribers to the other Seybold publication, **The Seybold Report**, is U.S. $54 per year. Outside the U.S. and Canada, please add $9 per year for airmail postage ($4.80 for each additional copy to be included in same envelope). **Checks from Canada and elsewhere outside the U.S. should be made payable in U.S. dollars directly to our bank:** *First Pennsylvania Bank, N.A., Philadelphia, Penna., 19101,* **into account No. 831-620-0.** *Please be sure to identify the name of subscriber if funds are transferred bank-to-bank.*

Company Name _____ *Attention of* _____

Address _____ *City, State and Zip* _____

☐ *Check enclosed.* ***Seybold Publications, Inc., Box 644, Media PA 19063*** ☐ *Please bill.*